# 基于大数据个性化音乐推荐算法分析

**摘　要：**音乐推荐算法针对当今时代信息过载的问题为用户推荐音乐的算法。音乐推荐具有物品空间大、用户空间大、物品消费代价小、物品重复使用率高、物品需求量大、物品喜好变化快、社会化程度高等特点。当今音乐推荐算法主要依赖于用户对音乐的操作记录建立用户和音乐的相似性矩阵，进行基于用户的协同过滤推荐或者基于物品的协同过滤；本文在音乐维度和用户维度之外，增加了对操作维度的分析，分析用户对音乐行为的产生的操作源，借此预测用户对音乐的喜好性；对于各个维度特征的筛选采用稳定性选择（Stability Selection）中的随机逻辑回归获得各个特征的分数，并将筛选后的特征通过 Light Gradient Boosting Tree（LightGBM）建立模型进行分析。此外本文创新使用数值特征地理进行预测，经验证，使用数值特征进行建模在有效性和鲁棒性上都有稳定的提升，并且该分析方法适用于所有有监督学习的分类学习，具有广泛的应用意义。

**关键词：**音乐推荐；数值特征；随机逻辑回归模型；LightGBM

**Abstract:** The music recommendation algorithm recommends the music algorithm for the user in view of the problem of information overload in the modern era. The music recommendation has features such as large item space, large user space, low product cost, high reusability of items, large demand for items, rapid change in item preferences, and high degree of socialization. Today's music recommendation algorithm mainly relies on the user's operation record of music to establish a similarity matrix between user and music, and performs user-based collaborative filtering recommendation or collaborative filtering based on items; this article adds operations to the music dimensions and user dimensions. Dimensional analysis, analysis of the user's operation source for the generation of music behavior, to predict the user's preference for music; for the screening of each dimension feature, the stochastic logistic regression in stability selection (Stability Selection) is used to obtain the score of each feature. The selected features were analyzed using the Light Gradient Boosting Tree (LightGBM) model. In addition, this paper innovatively uses numerical feature geography for forecasting. It is verified that the use of numerical features for modeling has a steady improvement in both effectiveness and robustness, and this analysis method is applicable to all classed learning with supervised learning, with a wide range of Application .

**Keywords:** Music recommendation; numerical characteristics; stochastic logistic regression model; LightGBM

## 一、前　　言

音乐推荐算法，就是针对音乐自身的内容特征以及用户的听歌行为，为广大用户提供可能符合他们兴趣爱好的歌曲的算法。而基于大数据的个性化音乐推荐算法，能够通过历史数据，别的用户的历史数据分析出潜在的喜好相似性，为用户更准确地挖掘出潜在的喜欢的音乐。

1995年，Ringo[1]算法的开发成就了历史上第一个推荐算法，可以向用户推荐他们喜欢的音乐并预测用户对特定音乐的评分，之后一段时间内，音乐推荐都是基于音乐曲目的基本信息产生，缺乏针对性。国外著名网站Pandora和Last.fm是最早提出音乐个性化推荐的网站。Pandora的音乐推荐算法主要来源于音乐基因工程（music gene）的项目[2]，根据这些基因计算歌曲的相似度，给用户推荐基因相似度高的音乐。国内也涌现了一些优秀的音乐推荐网站如豆瓣电台、虾米音乐、网易云音乐等等，根据用户平时推荐给好友的歌曲，听歌行为以及歌曲收录信息，找到“相似的品味者”，更好的做出推荐。

本文针对传统基于用户或者基于物品的协同过滤推荐方法在复杂场景下对用户进行音乐推荐占用内存大计算速度慢等缺点，提出一种基于LightGBM决策树算法的音乐推荐算法，使用相关性分析和稳定性选

择中的随机逻辑回归进行特征选择，采用数值特征取代个体特征进行用户对音乐的喜好预测，根据不同的候选集，可以形成不同推荐列表。采用kkbox音乐公司公布在Kaggle比赛平台上的用户、音乐、用户操作信息进行验证，预测准确率高达76%，训练时间9min，优于该比赛第一名用户算法的准确率68.4%。采用的算法模型可拓展性强，计算效率高，占用内存小，可以迁移到其它类型的推荐系统中。

## 二、推荐算法介绍

2.1 传统推荐算法

传统的推荐系统方法包括基于内容推荐过滤、基于规则的推荐、协同过滤推荐。

基于内容的过滤推荐根据物品的元数据，计算物品的相似性，然后基于用户的历史行为推荐给用户相似的物品；基于规则的推荐常使用于电子商务系统，大量的交易数据中获取关联规则或者按照时间购买商品的序列模型，进行物品之间的相互推荐；协同过滤包括基于用户的协同过滤和基于物品的协同过滤；基于用户的协同过滤通过分析用户历史行为，计算用户之间相似度，利用用户相似度和用户的历史行为给用户形成推荐列表。基于物品的协同过滤与之类似，分析用户行为计算物品之间的相似度，然后根据用户的历史偏好信息，将类似的物品推荐给用户。

2.2 基于 LightGBM 决策树模型的推荐算法

决策树算法的发展过程从 C3.0（基于信息增益）→CART（基于基尼系数）→提升树（AdaBoost）→梯度提升树（GDBT）→XGBosot → LightGBM 算法。

基于决策树模型的推荐算法具有以下优点：（1）可以并行化训练；（2）能够处理离散连续特征值和类别特征，不用对特征做归一化；（3）能够处理缺失值；（4）可以处理高维特征。

LightGBM（Light Gradient Boosting Machine）是 2017 年 8 月微软公司开源的基于决策树算法的分布式梯度提升框架，和之前的提升框架相比有更快的训练效率，更低的内存使用，更高的准确率，支持并行化学习，可以处理大规模数据等优点，可以用于排序，分类和许多其他机器学习任务。[3]

Boosting算法（提升法）指的是迭代算法，核心思想是对训练样本进行k次迭代，每次迭代形成一个弱学习器，然后根据学习误差对分类错误的样本加大训练权重，形成新的带有权重的训练集，训练形成新的弱学习器；最后将这些弱学习器根据结合策略形成一个强学习器，学习过程如图2.1所示：

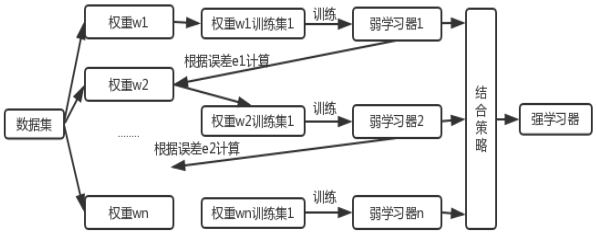


图2.1 Boosting算法学习过程

此外LightGBM利用Histogram的决策树算法，先把连续的浮点特征值离散化为k个整数，构造一个宽度为k的直方图，如图2.2所示，遍历数据时，根据离散化后的值作为索引在直方图中累积统计量，然后根据直方图的离散值，遍历寻找最优的分割点。使用直方图算法因为只保存特征离散化后的值，内存消耗可以降低为原来的1/8左右；此外计算的成本也大大降低，因为预排序算法每遍历一个特征值就需要计算一次分裂的增益，而直方图算法只用计算k（k为直方的个数），时间复杂度从从O(data*feature)优化到O(k*features)。

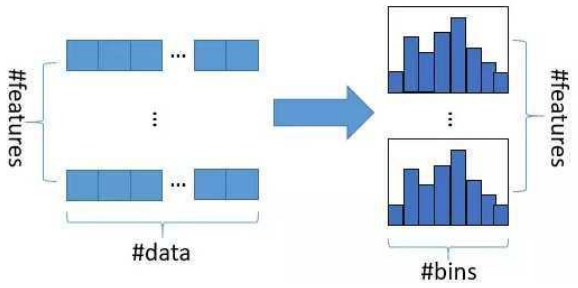


图2.2 直方图分割算法

和Xgboost采用level-wise策略相比，LightGBM采用更高效Leaf-wise策略（如图2.3所示），每次从当前所有叶子中，找到分裂增益最大的一个叶子，然后分类，如此循环，和Level-wise相比，分裂次数相同的情况下，可以降低更多的误差，获得更高的精度。同时LightGBM可以通过最大深度的限制防止过拟合。


图2.3　按层次生长策略（左）和按叶子次生长策略（右）

并且最新的LightGBM可以自动处理缺失值，可以进一步优化类别特征（Categorical Feature），不再使用类似one-hot coding的分割方式，对于类别数量很多的类别特征采用one-vs-other的切分方式长出的不均衡的树，采用many-vs-many的切分方式，寻找最优分割。

## 三、数据集以及数据预处理

3.1 数据集介绍

数据挖掘是在大量的、潜在有用的数据中挖掘出有用模式的过程。因此，原数据的质量直接影响到挖掘的效果，高质量的数据是进行有效挖掘的前提。

本文采用 kaggle 平台上 kkbox 举办的—KKBox's Music Recommendation Challenge 比赛的公开数据集，KKbox 是亚洲领先的音乐流媒体服务商，拥有全球最全面的亚洲流行音乐库，音乐曲目超过 3000 万首。官方声明比赛数据都来自都来自网页快照的抽样，除了对用户 id 进行了加密处理，其余数据都是原始数据[4]。

数据集共分为三个维度：用户维度，音乐维度，操作维度。

用户维度信息包括用户 ID、居住城市、年龄、性别、注册方式、注册时间；音乐维度信息包括歌曲 ID，歌曲长度，流派信息，艺人名字，作曲者，作词者，歌唱语言，歌名，ISRC 码；操作维度包括用户 ID，歌曲 ID，首次操作功能区，首次操作界面名，首次播放类型，首次收听一个月内是否重复完整收听。

在本文中，保留使用用户首次收听一个月内是否重复完整收听一首歌为评判用户喜好的标准。

3.2 数据筛选

在推荐系统中应该建立如下观点：操作次数特别少的用户和操作次数特别少的物品虽然占了绝大多数，但是这部分行为不具备统计规律，不能真实反映用户的喜好，选择这些数据进行训练，不能得到正确的结果。

本文根据实验分析得出，对于本文的数据集应该选择{播放次数大于 10 次的音乐的操作记录}∩{播放次数大于 35 次的用户操作记录}进行训练。

3.3 数据预处理

用户信息表有 21965 名用户的城市、性别、年龄、注册方式以及注册时间等信息。经查看，性别缺失率高达 47.45%，并且注册时用户填写性别也不一定真实，所以删除该特征。并且通过对年龄信息的查看，

年龄 0 岁的 10377 人，缺失值也达到了 50%，剩余年龄分布集中在 22-30 岁之间，区分度也不是很大。所以暂时也删除。注册时间的格式为%Y%M%D，全部转化为注册天数的连续变量。并且统计每个特征单个元素的播放次数和重复率，添加到用户信息表中。

音乐信息表中经筛选后的音乐只有一首歌缺失语言信息，经查验，该歌曲为 JONGHYUN 组合演唱的《White T-Shirt》，为韩文歌，我们进行人工填充，韩语对应的语言类别 31；缺失了 485 首歌曲的 genre_ids，缺失率为 1.6%，最频繁项为 465，出现频次为 16735，占据 50%的歌曲；因此对于缺失的少数 genre_ids，用最频繁项填充。并且 80%以上的歌曲只有一个 genre 类别，除了 1 首歌，其余歌曲最多两个类别，所以在本文保留两个 genre_ids，对于艺人名，作词者作曲者都不做缺失值填充,只进行 LabelEncoder 标签化处理。统计每个特征单个元素的播放次数和重复率，以及对应歌曲数，添加到音乐信息表单中作为新的数值特征。

用户对音乐的操作信息只出现在了用户操作表中，一共有三个特征，用户操作来源，用户操作界面布局，用户第一次听这首歌的来源，本文把把这三个特征归为操作维度一类。经统计首次操作功能区（9 种元素），首次操作界面名（20 种元素），首次播放类型（13 种元素）组成的子类别在所统计的操作共有 479 种组合，是 9*20*13=2340 的 1/5 左右。分别统计这 479 中组合的重复收听率，认为操作次数大于 20 的为有效统计子类别,子类别和 target 相关系数高，所以对操作维度分析采用子类别进行分析。

## 四、特征选择以及模型性能比较

4.1 音乐维度特征选择

音乐信息表中包含歌曲 ID，艺人 ID，作曲家 ID，作词者 ID，语言 ID，公司 ID，第一第二流派 ID，发布年份，以及这 9 个分类变量数值化的特征：播放次数和重复播放率，以及对应歌曲数目；歌曲长度，特征维度达到近 50 个，为了模型分析简单，而且避免过拟合，进行特征筛选。

统计这些分类特征对应歌曲数和播放次数的相关系数，绘制热力学分析图，如图 4.1 所示：

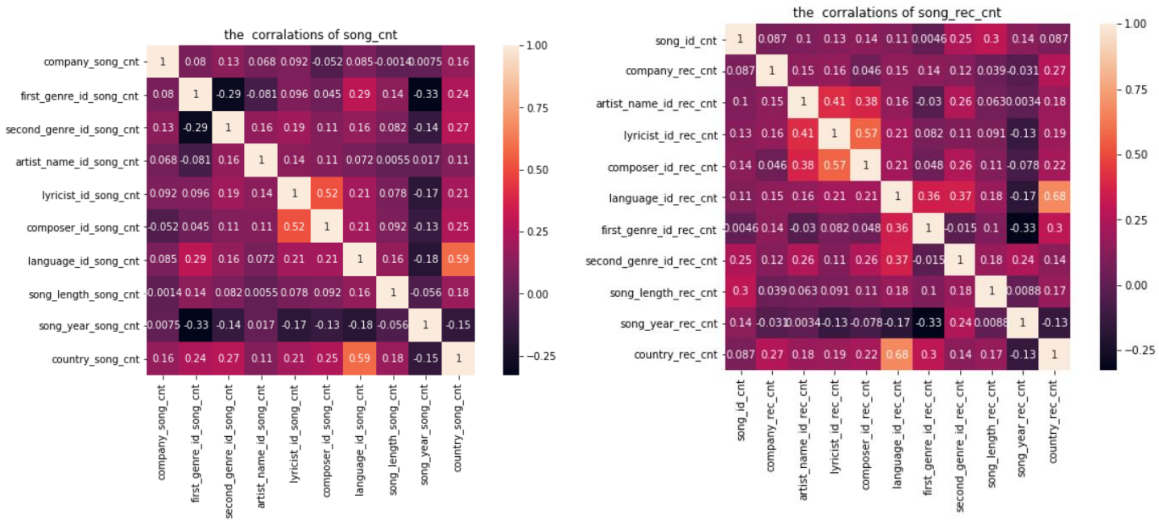


图 4.1 分类特征对应歌曲数（左）对应播放次数（右）相关系数热力学分布图

两幅图中 language 和 country 都体现了很强的相关性，这和经验认知国家和歌唱语言是强相关有关，并且国家信息提取自 ISRC 码，具有缺失，language 经过之前的缺失值补充没有缺失值，所以删除所有 country 有关的特征。

此外，lyricist 和 composer 以及 artist_name 之间也有强相关系数，但是无法直接比较数据质量，故采用稳定性回归中的随机逻辑回归对特征评分，评价各个分类特征的重复收听率对 target 重要性，对缺失的重复率用均值补充,评分如表 4.1 所示：

表 4.1 各分类特征重复率评分

| 分类变量 | member | song_id | artist | lyricist | language | composer |
|---|---|---|---|---|---|---|
| 第一次评分 | 1 | 1 | 1 | 1 | 1 | 0.07 |
| 第二次评分 | 1 | 1 | 1 | 1 | 1 | 0.06 |
| 分类变量 | company | first_genre | second_genr | song_length | song_year | |
| 第一次评分 | 0.02 | 0.8 | 0.39 | 1 | 0.665 | |
| 第二次评分 | 0.02 | 0.84 | 0.41 | 1 | 0.645 | |

由上表两次评分可以看出 composer 和是否重复播放无关，所以删除所有 composer 相关信息。对于 second_genre 因为缺失值严重，评分系数也不高，删除，company 可以看出和歌曲是否重复播放没有关系，故也删除。

4.2 特征性质选择

本文将使用 LightGBM 对全部使用数值特征进行训练预测，全部使用个体特征进行训练预测，以及使用全部特征（包括个体特征和数值特征）进行训练预测进行了详细的分析比较。

个体特征共 12 维，包括：用户，歌曲，操作类型，居住城市，注册方式，注册时间，歌曲长度，歌曲年份，歌曲首个流派属性，艺人，作词者，歌唱语言；数值特征共 30 维，包括：用户维度操作维度对应的操作次数和重复播放率，音乐维度对应的操作次数，重复播放率和对应音乐数。 全部特征及上述个体特征和数值特征相加，共 40 维。

（1）使用同样最大树深的预测准确率比较

因为个体特征一共 12 维，所以将共同树深设置为 12 进行比较，在训练集和测试集中的训练集结果如图 4.2 所示，可知在训练集中使用数值特征和全部特征进行预测的性质远优于使用个体特征进行预测，并且使用全部特征进行预测略优于使用数值特征进行预测；但是在测试集中，数值特征预测准确率有 2%左右的下降，使用个体特征也有 1%的下降，并且预测准确率远低于使用数值特征进行预测，使用全部特征的预测随着迭代次数的增加预测准确率一直下降，说明模型出现了过拟合现象。

（2）使用特征维度对应的树深的预测准确率比较

根据特征维度的不同，使用 LightGBM 训练时设置的最大树深也不同，全部使用个体特征的最大树深为 12，全部使用数值特征的最大树深为 30，使用全部特征的最大树深为 42，在训练集和测试集中的训练集结果如图 4.3 所示。

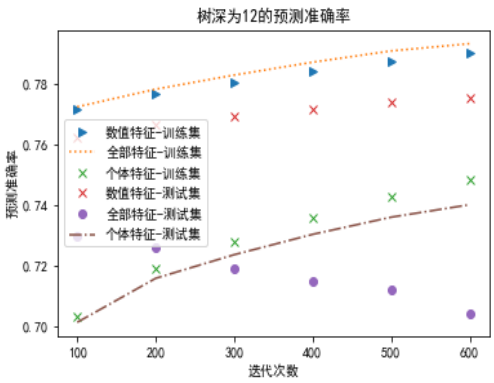


图 4.2 树深为 12 的预测准确率

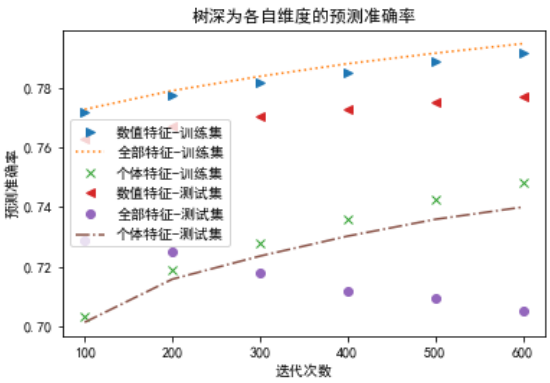


图 4.3 树深为各自维度的预测准确率

观察图 4.3 可知，当最大树深为各自特征维度时，总体趋势和使用最大树深为 12 的相同，没有模型出现明显的性能改变。

（3）所有 LightGBM 模型比较

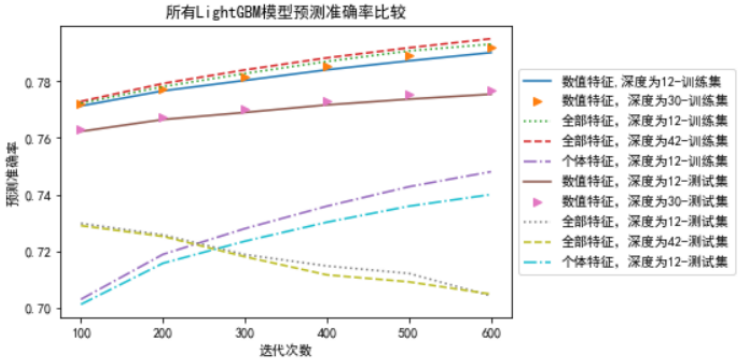


图 4.4 所有 LightGBM 模型预测准确率比较

图 4.4 中比较了所有 LightGBM 模型在训练集和测试集的预测准确率比较，可以得知在训练集中使用全部特征进行训练时，使用特征维度的最大树深 42 的预测准确率最高，但是和使用全部特征训练树深 12 和使用数值特征进行训练的两个模型相比，性能差距不大。但是在测试集中，使用数值特征进行训练的模型预测准确率最高，远优于其他 LGBM 模型，并且树深为 30 的模型略优于树深为 12 的模型。

使用个体特征训练的模型随着迭代次数的增加，预测准确率一直提升，但是还是远低于使用数值特征训练的模型；使用全部特征训练的模型，由于过拟合的问题，随着迭代次数的增加，预测准确率越来越低。

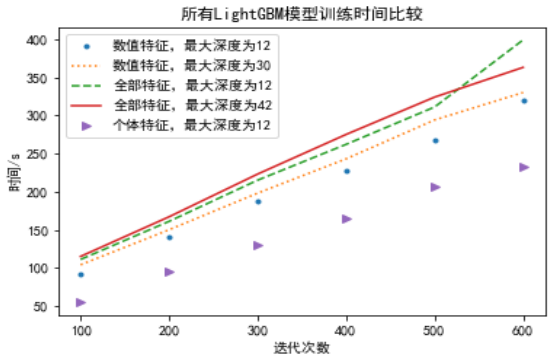


图 4.5 所有 LightGBM 模型训练时间比较

由图 4.5 可知使用个体特征进行训练的时间成本最小，使用数值特征和使用全部特征进行训练的时间成本差距不大。总体来说，使用全部特征的模型时间成本略大于全部使用数值特征的模型；最大深度大的模型的时间成本大于最大深度小的模型。

综合上述性能比较可知，使用数值特征训练模型性能远优于传统使用个体特征进行训练或者使用全部特征进行训练的模型，预测准确率在训练集和测试集差距不大，训练集中的预测准确率 78%以上，测试集预测准确率 76%以上，并且随着迭代次数一直增加。

本文还使用逻辑回归模型使用数值特征进行了训练，将训练后二元化结果和全部使用个体特征、全部使用数值特征、使用全部特征的 LightGBM 模型（最大树深为 10，迭代次数 150 次）进行了比较，将将混淆矩阵信息列入表 4.2，4.3：

表 4.2 训练集不同算法混淆矩阵比较（预测值/实际值）

| 算法 | 1/1 | 0/1 | 1/0 | 0/0 | 平均正确率 |
|---|---|---|---|---|---|
| 逻辑回归 | 1432368 | 840961 | 698144 | 1896378 | 0.68382249 |
| 用数值特征的 lgbm | 1629416 | 643913 | 813643 | 1780879 | 0.70057506 |
| 用分类特征的 lgbm | 1407061 | 866268 | 817165 | 1777357 | 0.65417327 |
| 用全部特征的 lgbm | 1631504 | 641825 | 811447 | 1783075 | 0.70145512 |

表 4.3 测试集不同算法混淆矩阵比较（预测值/实际值）

| 算法 | 1/1 | 0/1 | 1/0 | 0/0 | 平均正确率 |
|---|---|---|---|---|---|
| 逻辑回归 | 353976 | 214034 | 179268 | 469685 | 0.67681680 |
| 用数值特征的 lgbm | 403227 | 164783 | 208494 | 440459 | 0.69327169 |
| 用分类特征的 lgbm | 350684 | 217326 | 205062 | 443891 | 0.65291632 |
| 用全部特征的 lgbm | 407394 | 160616 | 255973 | 392980 | 0.65768146 |

运行时间比较：logitic_regression 训练时间 47.8s，用分类特征的 lgbm 训练时间 1min5s，用数值特征的训练时间 1min39s，用所有特征的训练之间为 2min20s，对于数值特征使用简单的逻辑回归模型对结果预测，无论在训练集还是在测试集上的预测正确率都优于使用更复杂模型的使用分类特征的预测正确率，甚至在测试集上也优于使用了全部特征训练的 lgbm 模型。说明在有监督的模型中，对于分类特征的训练，可以使用数值特征替代，极有可能获得更优秀的训练效果。

4.3 迭代次数选择

使用数值特征训练 LightGBM 模型在训练集和测试集中随着迭代次数的变化预测准确率的变化如图 4.8 所示，模型使用为最大树深为 12 的使用数值特征进行训练的 LightGBM 模型，使用二元化后的数值计算准确率：

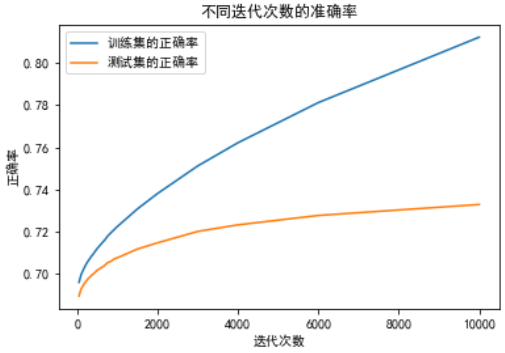


图 4.8 使用数值特征不同迭代次数下的预测准确率曲线

由图可知，由于 feature_fraction 和 bagging_fraction 的参数设置，并且使用的是数值特征，增加迭代次数也没有导致过拟合的问题，预测准确率一直提升。但是需要注意，当迭代次数大于 3000 次之后，在训练集中的预测准确率虽然仍在快速提升，但是在测试集中的预测准确率提升非常细微。由之前是实验可知，模型训练成本随着迭代次数的增加线性增加，所以盲目增加迭代次数换取细微的增益再实际应用中是得不偿失的，所以在实际使用中，音乐服务商应该考虑实际需求设置迭代次数，本文作者建议在 3000 次以下。

4.3 其它模型参数设置

学习控制参数有：min_data_in_leaf，一个叶子上数据最小的数量，用来处理过拟合，默认为 20；feature_fraction，默认为 1.0，取值范围 0~1，如果取值小于 1，LightGBM 将会在每次迭代中随机选择部分特征，可以加速训练，也可以用来处理过拟合，本文设置为 0.8。feature_fraction_seed，feature_fraction 的随机数种子，默认为 2；bagging_fraction,默认为，和 feature_fraction 功能类似，但是在不进行重新采样的情况下随机选择部分数据，设置为 0.9；bagging_freq，默认为 0，但是要使用 bagging_fraction 必须为非零值，k 意味着每 k 次迭代执行 bagging，本文设置为 2；bagging_seed，bagging 随机数种子，默认为 3；max_depth，用来限制树模型的最大深度，默认为-1，意味着没有限制，本文设置为 10； categorical_feature 用来指定分类特征。using_missing，设置为 False，禁用缺失值。max_bin，默认 255，LightGBM 使用 unit8 压缩内存，所以本文设置为 256，使用 unit16 压缩内存；verbosity 默认为 1，设置为 0，只输出警告信息。

## 五、实验结果以及分析

本文对筛选后的 train 数据集使用 Sklearn 库中的 split 函数随机将其分割成 5 个不交叉子集，每次选择 4 个作为新训练集和 1 个作为新测试集，比例为 80%和 20%。根据新训练中的数据重新计算数据集中的数值特征,并且对播放次数进行 z_score 归一化，并且将计算的数值特征应用到新测试集中。

本文采用 ROC 曲线下的面积来衡量预测准确率，采用模型训练时间衡量模型的有效性。

5.1 二元化预测结果比较

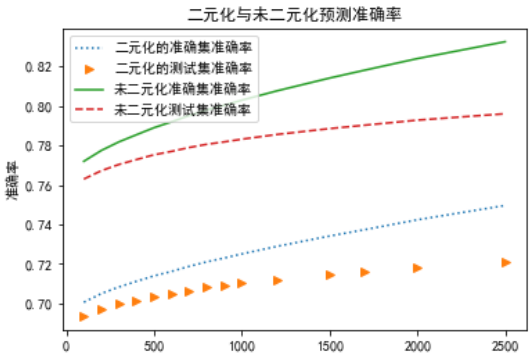


图 5.1 二元化与未二元化预测准确率

因为 kkbox 在 kaggle 平台比赛要求提交结果为二元化结果进行准确率评分，所以本文将使用数值特征训练的 LightGBM 模型，最大树深为 30 层的预测结果二元化，计算预测 ROC 评分。如图 5.1 所示，和未二元化的预测结果相比，预测准确率下降。

在训练集中二元化后的测试准确率在迭代次数 1000 次时到达 0.72，测试集中二元化测试准确率大于 0.70，该比赛第一名的预测概率为 0.684。并且根据之前的实验可以得知，使用该模型迭代 1000 次的平均耗费时间为 9min15s，远小于其它模型的训练时间。

5.2 不使用操作特征进行预测

本文将输入操作特征最大树深为 30 的模型和不输入操作特征最大树深为 28 的模型进行比较。

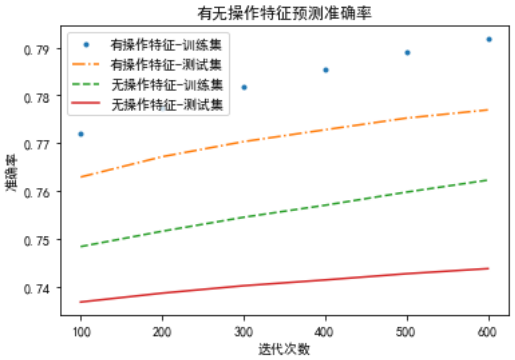


图 5.2 有无操作特征预测准确率比较

由图 5.2 可知，有操作特征的预测准确率明显高于无操作特征的预测准确率。但在没有操作特征时，即对于用户未听过的音乐，预测准确率在训练集在迭代次数 500 次时已经达到 75%，测试集上达到 74%，能够满足音乐服务商的应用需求，并且平均时间成本只需要 3min15s，所以可以采用该模型对用户没有听过的音乐进行预测。

5.3 使用预测结果进行推荐

本文随机选择一次测试集和训练集结果进行推荐结果展示，给用户的推荐列表应该包括用户听过的歌

曲和用户没听过的歌曲。

使用数值特征，树深 30，迭代次数 600 的模型进行预测。选择三个用户（用户信息如表 5.1 所示）进行结果展示，序号分别为对应信息如下：

表 5.1　用户信息表

| 用户序号 | 播放次数 | 喜欢艺人 | 喜欢流派 | 城市 | 注册方式 | 注册时间 |
|---|---|---|---|---|---|---|
| 6408 | 311 | 谢和弦(17) | 通俗流行(169) | 0 | 2 | 15525 |
| 11143 | 933 | 陈奕迅(43) | 通俗流行(420) | 11 | 3 | 15842 |
| 782 | 738 | V\| Jin (35) | 通俗流行(427) | 0 | 0 | 16148 |

以 11143 用户举例，以陈奕迅的所有音乐为候选集，然后进行喜好预测，用没有操作类型的模型，经过筛选后的陈奕迅的歌曲有 193 首。经过预测之后选择喜欢概率高的形成推荐列表，如表 5.2 所示。

表 2　用户 11163 由喜欢的歌手生成的推荐列表

| 歌名 | 预测重复率 |
|---|---|
| 讓我留在你身邊 | 0.811024012 |
| 淘汰(國) | 0.692591114 |
| 愛情轉移(國) | 0.667852494 |
| 可以了 | 0.655677032 |
| 陰天快樂 | 0.627131943 |

对于用户 6408 、11143、782，喜欢通俗流行音乐，该流派音乐共计 17544 首，分别计算推荐前 5 名的歌曲，如表 5.3 所示。

表 5.3　不同用户由喜欢的相同的流派的推荐列表

| 6408 | | 11143 | | 782 | |
|---|---|---|---|---|---|
| 歌名 | 重复率 | 歌名 | 重复率 | 歌名 | 重复率 |
| 謝謝妳愛我 | 0.763 | 謝謝妳愛我 | 0.816 | 謝謝妳愛我 | 0.814 |
| 演員 | 0.753 | 演員 | 0.793 | 演員 | 0.813 |
| 小幸運 | 0.737 | 好愛好散 | 0.776 | FLY OUT | 0.795 |
| FLY OUT | 0.730 | 小幸運 | 0.770 | 犯錯 | 0.794 |
| 好愛好散 | 0.728 | FLY OUT | 0.763 | 不為誰而作的歌 | 0.788 |

由表 5.2 可知，对于同一个用户，设置不同的候选集，可以有不同的推荐列表，支持音乐服务商使用多种推荐方式，也可以综合多个候选集，建立混合推荐的推荐列表。

由表 5.3 可知，对于不同用户，因为其它用户特征的不同，对于相同的候选集，也可以生成不同的推荐列表，符合个性化推荐的要求。

## 六、结　　语

本文采用新的ligthgbm算法对用户是否会在一个月内重复收听某一首歌曲进行预测，以此作为个性化推荐的目标。通过分析数据特征，使用相关性以及稳定性选择等方法选择特征，随后通过对训练输入数值特征，分类特征和全部特征的性能比较，创新性选择用数值特征完全取代分类特征去训练模型进行预测，

使得模型在有效性和准确性上都有稳定的提升，对于其余需要再分类特征上建模的实验具有参考意义。

## 注释：

[1] Shardanand U. Social information filtering: algorithms for automating “word of mouth”[C]// Sigchi Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co. 1995:210-217.

[2]Tzanetakis G, Cook P. Musical genre classification of audio signals[J]. IEEE Transactions on Speech & Audio Processing, 2002, 10(5):293-302.

[3]https://github.com/Microsoft/LightGBM

[4]Kaggle，kkox-music-recommodation-challenge 数据介绍

https://www.kaggle.com/c/kkbox-music-recommendation-challenge/data

## 参考文献：


[1]Resnick P, Varian H R. Recommender systems[M]. ACM, 1997.

[2]Shardanand U. Social information filtering: algorithms for automating “word of mouth”[C]// Sigchi Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co. 1995:210-217.

[3]王中原. 面向互联网基于相关性挖掘的音乐推荐[D]. 浙江大学计算机科学与技术学院 浙江大学, 2008.

[4] Tzanetakis G, Cook P. Musical genre classification of audio signals[J]. IEEE Transactions on Speech & Audio Processing, 2002, 10(5):293-302.

[5] 刘建国, 周涛, 汪秉宏. 个性化推荐算法的研究进展[J]. 自然科学进展, 2009, 19(1):1-15.

[6]许海玲, 吴潇, 李晓东,等. 互联网推荐算法比较研究[J]. 软件学报, 2009, 20(2):350-362.

[7]Igel C, Suttorp T, Hansen N. Steady-State Selection and Efficient Covariance Matrix Update in the Multi-objective CMA-ES[C]// International Conference on Evolutionary Multi-Criterion Optimization. Springer-Verlag, 2007:171-185.

[8]微软 LightGBM 开源项目：https://github.com/Microsoft/LightGBM

[9]Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In Advances in Neural Information Processing Systems (NIPS), pp. 3149-3157. 2017.

[10]Qi Meng, Guolin Ke, Taifeng Wang, Wei Chen, Qiwei Ye, Zhi-Ming Ma, Tieyan Liu. "A Communication-Efficient Parallel Algorithm for Decision Tree". Advances in Neural Information Processing Systems 29 (NIPS 2016).

[11] Last.fm 数据集介绍:

https://labrosa.ee.columbia.edu/millionsong/lastfm

[12] 阿里流行音乐趋势预测大赛介绍:

https://tianchi.aliyun.com/competition/information.htm?spm=5176.11165320.5678.2.7bc4a0737qKl9G&raceId=231531

[13] 袁梅宇. 数控挖掘与机器学习:WEKA 应用技术与实践[M]. 清华大学出版社, 2016.

[14] 项亮. 推荐算法实践[M]. 人民邮电出版社, 2012.

[15] 张良均 … [等. Python 数据分析与挖掘实战[M]. 机械工业出版社, 2016.

[16] 米尔顿李芳. 深入浅出数据分析 : Head first data analysis[M]. 电子工业出版社, 2012.